Create simple overlay with UserControl in Windows Phone

This article explains how to create a simple overlay using `UserControl` in Windows Phone.

Introduction

Overlays can have a wide range of applications, be it a simple prompt or displaying a progress indicator. Using UserControl for implementing it gives us feasibility to easily construct any type of overlay. This article demonstrates how to implement a sample progress overlay.

The following steps describe how to implement it:

1. Create new Windows Phone project from Visual Studio.
2. Add new `UserControl` named **OverLay.xaml** in current project.

Copy and paste following code in your **OverLay.xaml** file.

```
<UserControl x:Class="WindowsPhoneSample7.OverLay"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    d:DesignHeight="800" d:DesignWidth="480">

    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.RowDefinitions>
            <RowDefinition Height="400"/>
            <RowDefinition Height="400"/>
        </Grid.RowDefinitions>
        <StackPanel Grid.Row="1">
            <ProgressBar IsIndeterminate="True" Foreground="Orange" Height="50"
Width="480" VerticalAlignment="Center"/>
            <TextBlock Text="Wait" Foreground="Orange" HorizontalAlignment="Center"/>
        </StackPanel>
    </Grid>
</UserControl>
```

`LayoutRoot` is populated with ProgressBar and TextBlock but you can put any UI controls here that you want to show on the overlay. For example, if you want to show a simple About prompt then you may put a `TextBlock` and an `Image` with custom size.

In code behind file of **Overlay.xaml** initialize the size of `LayoutRoot` to the value you want. The following code shows how the height and width of `LayoutRoot` are set to the height and width of the screen.

```
public OverLay()
{
     InitializeComponent();
     this.LayoutRoot.Height = Application.Current.Host.Content.ActualHeight;
     this.LayoutRoot.Width = Application.Current.Host.Content.ActualWidth;
     SystemTray.IsVisible = false; //to hide system tray
}
```


(C) Copyright Nokia 2014. All rights reserved.

Now open **MainPage.xaml.cs** file and copy and paste the following code in it:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;
using System.Windows.Controls.Primitives;
using System.ComponentModel;
using System.Threading;

namespace WindowsPhoneSample7
{
    public partial class MainPage : PhoneApplicationPage
    {
        private Popup popup;
        // Constructor
        public MainPage()
        {
            InitializeComponent();
            this.popup = new Popup();
        }

        private void BtnStart_Click(object sender, RoutedEventArgs e)
        {
            this.LayoutRoot.Opacity = 0.2;
            OverLay ovr = new OverLay();
            this.popup.Child = ovr;
            this.popup.IsOpen = true;
            BackgroundWorker worker = new BackgroundWorker();
            worker.DoWork += (s, a) =>
            {
                Thread.Sleep(5000);
            };
            worker.RunWorkerCompleted += (s, a) =>
            {
                popup.IsOpen = false;
                this.LayoutRoot.Opacity = 1.0;
            };
            worker.RunWorkerAsync();
        }

        protected override void OnBackKeyPress(System.ComponentModel.CancelEventArgs e)
        {
            this.popup.IsOpen = false;
        }
    }
}
```


(C) Copyright Nokia 2014. All rights reserved.

The `BtnStart_Click` event contains the logic of showing the progress overlay; when activating and deactivating the overlay you need to set the opacity of underlying page. In the DoWork method of BackgroundWorker, you can write the actual logic. In the current implementation, it just blocks the UI thread for some time.
The following screen shot shows the above implemententation of the progress overlay:

Progress Overlay

Finally don't forget to set `IsOpen` property of `popup` to **false** on `OnBackKeyPress` event, otherwise the overlay will continue to display even if you navigate to another XAML page.
You can find the complete source code below.
File:OverLays.zip

(C) Copyright Nokia 2014. All rights reserved.